Amendments to the Specification:

RELATED APPLICATION

[0001~~0~~0000] This patent application claims priority of U.S. Provisional Application Serial No. 60/448,508 filed February 19, 2003 titled "Method and Apparatus for Operations on Token Sequences: Computing Similarity, Extracting Term Values, and Searching Efficiently", which is hereby incorporated herein by reference.

[0082] In another embodiment of the invention the approach here may be used to solve this search problem in substantially M*log(N) time, instead of the M*N time of the straightforward approach. This approach involves pre-computing an indexing structure around the profiles and proceeds thusly:

1. Start with N text blocks.
2. Given a target text block T. The goal is to find the closest text block of T among the N text blocks.
3. Concatenate the N blocks to form a single text block.
4. Compute the profile of the single block.
5. Obtain the C <= N eigenvalues from the profile, and sort them by their magnitude. Call them w.0, ..., w.C-1, where w.0 is the eigenvalue with the largest magnitude.
6. For each of the N text blocks t.i, compute the transition probability matrix, b.i.

7. Starting from the largest eigenvalue w.0, compute the partial sum, s.i for each text block.

s.i = u.transpose.0 * b.i * v.0

8. Each of the s.i is a complex number in the complex plane. Partition the complex plane down the imaginary axis, the half with non-negative real part and the other half with negative real part. Each of the s.i will fall on one side or the other. Count the fraction f that falls on the non-negative real part side. One may say that there is a sufficiently good partitioning of the n points, if the fraction f falls within a predetermined range [0.3, 0.7] for example. If there is a good partitioning, then select left and right eigenvalue pair corresponding to w.0 as the "partitioning component."
9. If w.0 does not yield a good partitioning, then proceed to the eigenvector with the next largest magnitude. Do this iteratively until the k-th left/right eigenvector pair that yields a good partition is found. This is the partitioning component.
10. In practice, there will always be a largest component k. This follows from the properties of how the principal component analysis technique chooses the eigenvectors in the first place, or this follows from the definition of the left and right eigenvectors themselves.
11. Once the partitioning component is determined, use it to partition the n text blocks into the each respective half-plane.

12. Take each subset of text blocks and repeat the steps ~~Error! Reference source not found. Error! Reference source not found.~~3-11 above.

13. Repeat this procedure on each subset until each subset contains only one text block. In essence the final partition computes the specific profile for that sole text block T.